Knowledge Graph Enhanced Retrieval-Augmented Generation for Failure Mode and Effects Analysis

Lukas Bahr[1,3*], Christoph Wehner[2], Judith Wewerka[1], José Bittencourt[1], Ute Schmid[2], Rüdiger Daub[3,4]

[1]Digitization Department for Battery Production, BMW Group.
[2]Cognitive Systems Group, University of Bamberg.
[3]Institute for Machine Tools and Industrial Management, Technical University of Munich.
[4]Institute for Casting, Composite and Processing Technology (IGCV), Fraunhofer.

*Corresponding author(s). E-mail(s): lukas.bahr@bmw.de;

Abstract

Failure mode and effects analysis (FMEA) is a critical tool for mitigating potential failures, particular during ramp-up phases of new products. However, its effectiveness is often limited by the missing reasoning capabilities of the FMEA tools, which are usually tabular structured. Meanwhile, large language models (LLMs) offer novel prospects for fine-tuning on custom datasets for reasoning within FMEA contexts. However, LLMs face challenges in tasks that require factual knowledge, a gap that retrieval-augmented generation (RAG) approaches aim to fill. RAG retrieves information from a non-parametric data store and uses a language model to generate responses. Building on this idea, we propose to advance the non-parametric data store with a knowledge graph (KG). By enhancing the RAG framework with a KG, our objective is to leverage analytical and semantic question-answering capabilities on FMEA data. This paper contributes by presenting a new ontology for FMEA observations, an algorithm for creating vector embeddings from the FMEA KG, and a KG enhanced RAG framework. Our approach is validated through a human study and we measure the performance of the context retrieval recall and precision.

Keywords: FMEA, Risk Assessment, Knowledge Graph, Retrieval-Augmented Generation, Large Language Models

1 Introduction

A global and close-to-simultaneous start of production for a new product challenges multi- and interdisciplinary teams along the vertical and horizontal value chain. Quality assurance teams are in particular concerned during the start of production with poorly controlled processes [1]. Furthermore, processes are often complex and cause-and-effect relationships are difficult to identify, for example, due to a variety of operations, coordination, or communication barriers of a global production network [2, 3]. For this reason, experience gained in pre-series with regard to possible problems during the ramp-up and operation of production systems must be transferred to the series plants in a global production network.

Failure mode and effects analysis (FMEA) is one tool to avoid potential failures during a ramp-up phase. FMEA is a risk analysis tool that focuses on systematic identification of failures and the prevention of defects, for example, in the process chain or in the design of the product [4, 5]. Typically, many stakeholders are involved in complex products, leading to different and incoherent FMEA approaches that make it difficult to reason over the FMEA analysis between the different development units [6]. Furthermore, FMEA documents become quickly inextricable as actors maintain inconsistent data entries, leading to documentation that is difficult to evaluate with unclear completeness and integrity [7]. Lastly, FMEA results are usually tabular structured and miss reasoning capabilities, such as finding generalizations in error patterns, summarizing, or analytic guidance for error prevention.

Recent advances in large language models (LLMs) enable fine-tuning on custom data sets. This can be used for information retrieval and interactions with facts of the FMEA. LLMs are commonly trained on large amounts of text and show great potential, among others, in tasks such as language comprehension or answering questions [8]. However, LLMs often fail when factual knowledge is required [9]. Designing LLMs for tasks that require factual knowledge remains an open research question. In [10], the authors show that retrieval-augmented generation (RAG) outperforms language models for question-answering (QA) tasks that require factual precision. The RAG architecture stores the information in a non-parametric store and utilizes a language model to generate the answer. However, current RAG approaches face difficulties in extracting analytical data due to the loss of symbolic meaning attached to numerical values. This limits drawing conclusions from the numerical information of the FMEA. For example, *"What failure cause has the highest risk priority number for the process step X?"* cannot correctly be inferred from the non-parametric data store.

One possible solution is to embed the non-parametric data store in a knowledge graph (KG). KGs store symbolic facts by representing data as nodes and relations in a directed, labeled multi-graph [11]. The symbolic structure of the KG contextualizes the data and allows interpretable reasoners to derive novel insights from the data [12]. Prominent tools such as search engines (Google, Bing, etc.) or query answering services (Apple Siri, Amazon Alexa, etc.) have recognized the benefits of KGs [13, 14].

This paper contributes by integrating KGs within a RAG framework for QA on FMEA data. The paper proposes modeling the FMEA observations as a property KG and, therefore, introduces a novel ontology for FMEA. Subsequently, it suggests an algorithm for traversing the FMEA-KG to create vector embeddings. Finally, the paper

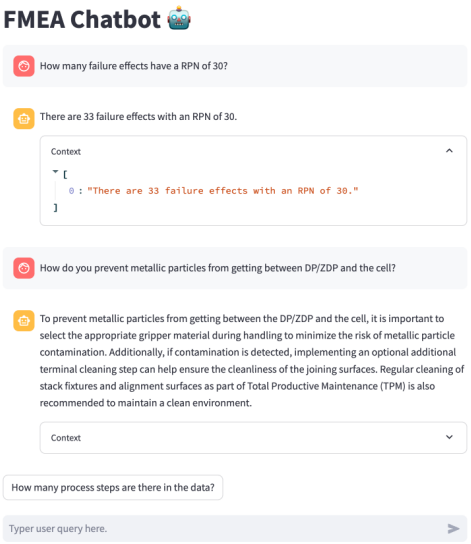

Fig. 1 Screen capture of the FMEA chatbot. The user interface is kept simple. Clicking on context reveals the information retrieved from the FMEA-KG.

demonstrates how to integrate the non-parametric data store into a KG, resulting in a KG-enhanced RAG (KG RAG) framework. The KG RAG framework enables the retrieval of hallucination-free and human-understandable information while allowing analytical capabilities through the KG graph query. A human study provides information on the effectiveness of the approach and measures the performance of context recall, precision, and user-friendliness.

The paper is organized as follows. In Section 2, we recall the main building steps that enable our KG RAG framework. In Section 3, we detail the ontology for the FMEA observations, illustrate the embedding steps for creating the vector index nodes, and propose our KG RAG framework. Finally, we illustrate the merits of the approach through a human study and context retrieval evaluation in Section 4, before concluding in Section 5.

2 Related Work

The following briefly introduces the influential work on FMEA, KG, LLM, and RAG relevant to this paper. The methods defined in ISO/IEC 31010 [4] dominate the risk assessment in manufacturing: the five whys method [15], fault tree analysis [16], and failure mode and effect analysis (FMEA) [17]. FMEA establishes a common framework without relying on statistics, making it an efficient tool for tracking potential product and process failures [5]. Current trends in FMEA can be found in [18, 19]. However, challenges arise when dealing with systems, such as in manufacturing [20]. In particular, incoherent FMEA approaches remain due to the intersection between different domains [6], difficulties in identifying and managing errors [21], and large and poorly

structured FMEA documents [7]. There are methods that aim to facilitate the accessibility of FMEA documents, such as integrating it with quality function deployment [22].

Instead of storing the FMEA data in a classical relational (or non-relational) database, KGs offer versatile access to information. KGs are a way to store and retrieve facts [11], making them a pivotal technology for answering questions based on factual knowledge [10]. Furthermore, novel KG tools allow the storage of vector embeddings [23]. In [24], KGs are initially conceived as semantic networks and have diverse applications, including ventures into human language through projects such as WordNet [25]. Subsequently, a multitude of private companies and academic institutions have embraced KGs for a wide array of applications [11], such as DBpedia [26], Freebase [27], or Google KG [28]. The default method of interacting with formalized knowledge in KGs is a structured query language (e.g., Cypher, SPARQL)[11, 29] , which requires domain knowledge of the KG schema for effective data extraction, making interactions cumbersome for non-domain experts.

Advances in LLMs allow for fine-tuning on custom data sets, enabling a retrieval approach for FMEA observations. The LLM's transformer architecture enables training with its self-attention mechanism for multiple sequences that simplify, for example, the optimization process or minimize the risk of vanishing gradients [8, 30]. LLMs are trained on large amounts of data and tackle various tasks in the field of natural language processing (NLP), such as language comprehension or QA, and have also found adoption in the manufacturing domain [8, 31]. However, LLMs often fail in tasks in which factual knowledge is required [9], making LLMs unreliable in expert domains critical to manufacturing. Furthermore, LLMs tend to hallucinate, leading to misinformation and making them unreliable in tasks where accurate knowledge recall is crucial [32, 33]. Additionally, LLMs are a black-box architecture that result in a lack of interpretability and complicates understanding of the reasoning of the model [34, 35].

Rather than fine-tuning a black-box model on FMEA data, LLMs allow retrieving vector embedding of its input, permitting architectures for better explainable models such as RAG [36]. RAG approaches save observations (strings) as vector embeddings in a non-parametric store and utilize natural language generation methods to generate the answer from the store [37]. This approach allows for the composition of responses that minimize hallucinations, making the responses of chat-based LLMs more fact-based. However, since the information is represented as a vector, the literal symbolic meaning of the numerical values is lost, which remains a challenge for current RAG approaches [38]. For example, retrieving the highest risk priority number of the FMEA likely outputs the wrong information, as the symbolic meaning of the number is embedded in a vector.

Vector embeddings represent the strings in a high-dimensional vector space. In [39], the authors initially propose the embedding of words into a meaningful representation by taking advantage of the vector dimensionalities. Novel advanced methods, such as BERT [40], adopt transformer-based self-supervised language models, resulting in richer representations of the language structure. In [41, 42], it is shown that pre-training on a sufficiently large batch size can lead to high-quality vector representations.

Approaches such as GreaseML [43] or UniKGQA [44] aim to link and reason over the KG space using natural language, while providing access to the latest knowledge without retraining. Another research stream on risk assessment analysis with LLM focuses on retraining the LLM, for example, root cause analysis for recommending cloud incidents [45]. Although QA with KG has already been proposed in other domains, such as electric power generation [46], it is yet to be explored to utilize RAG for QA in the manufacturing domain to allow interaction with FMEA data. This paper aims to address this gap and simplify interactions for non-domain experts.

3 QA for FMEA

In the following, we introduce the KG RAG framework that allows QA for FMEA data. The framework takes advantage of recent developments in the field of LLM. In contrast to fine-tuning a LLM on the FMEA data basis, the proposed KG RAG framework enables (i) dynamic data updating without relearning and (ii) basic numerical analytics such as finding the failure cause with the highest risk priority number. In Section 3.1, we present an ontology designed to store FMEA data in the KG and outline an algorithm to compute vector embeddings from the FMEA KG. Following, in Section 3.2, we introduce a KG RAG framework that retrieves the FMEA context through the vector search, enhanced with the KG graph query, to deliver tailored results for the user inquiry.

3.1 Embodying of FMEA Data in a KG

We propose a two-step process to store and embed FMEA data in a KG. Typically, FMEA data are tabular structured in a relational (or non-relational) database. First, the tabular data structure must be transposed into a graph structure, for which the definition of an ontology is required. Second, to enable the graph structure for a vector search, the FMEA data have to be aggregated and embedded using a word-to-vector encoder. The resulting embeddings are attached to the KG as nodes.

Transposing the FMEA data into a KG. Consider a KG G to be a directed graph defined by

$$G = (E, R, F), \tag{1}$$

where $e \in E$ is the set of entities (or nodes) of the graph and $r \in R$ are the relations (or edges) of the graph. $F = (e_h, r, e_t)$ denotes a set of facts (triples) consisting of a head entity e_h and a tail entity e_t connected by the relation r [11]. KGs enhanced with literals add additional descriptive features to the graph. These literals, or features, provide extra information about entities and their relations. This type of KG is called a property graph and an entity is represented as a tuple $e = (s_e, l_1, ..., l_n)$, where s_e denotes the symbol of the entity e, such as its name, and l represents literals from 1 to n. Literals can be a descriptive string, a numeric value, or any other piece of information about the entity. Similarly, a relationship is indicated as $r = (s_r, l_1, ..., l_m)$, where s_r is the symbol that identifies the relationship r, and the literals 1 to m provide additional information about the connection, such as the date the relationship was

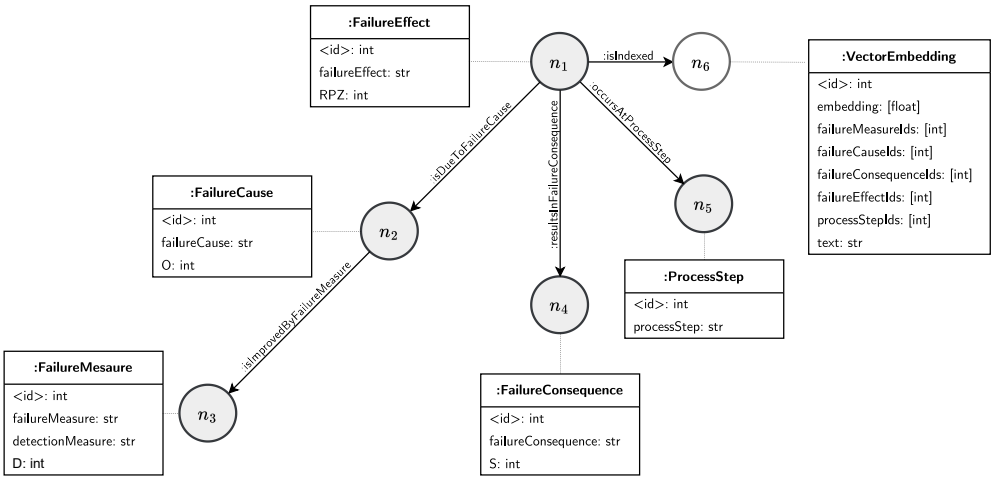

Fig. 2 Overview of the proposed ontology for the FMEA stating the literals and relations of the entities. The grey entities ($n_1 - n_5$) depict the information chunk that gets embedded as *VectorEmbedding* node.

established. n and m define the number of literals. It is important to note that the number of literals associated with any given entity or relation can vary.

KG can be structured using an ontology that provides a set of rules and classifications specific to the domain. This includes establishing a class hierarchy and defining the types of relation, such as their origin and target categories. For example, a *Child* entity is linked to a *Parent* entity through a *hasParent* relationship. Within the KG, we propose the following ontology, in which each node corresponds to its entity associated with the FMEA. The FMEA KG types are defined by *ProcessStep*, *FailureEffect*, *FailureConsequence*, *FailureCause*, and *FailureMeasure*. We represent *DetectionMeasure* and numerical variables such as *RiskPriorityNumber (RPN)*, *Severity (S)*, *Occurrence (O)*, and *Detection (D)* as literals of the respective entities. Figure 2 shows the complete ontology. Numerical values for the risk assessment of S, O and D range from 1 to 10 and describe the severity of the outcome for the specific FMEA entity. A higher number indicates a higher probability or severity of occurrence. The RPN is calculated by $RPN = S \cdot O \cdot D$ and provides a quantitative measure of the risk associated with each failure mode in the FMEA [5]. This helps in ranking the failure modes; the higher the score, the worse the risk.

The embodiment of the FMEA as KG allows the employment of (i) graph algorithms such as reasoning or path finding [11], and (ii) a query language that facilitates the retrieval of complex joint information. Although there are sophisticated reasoning methods within KGs, e.g., ruled-based or distributed representation-based inference methods, they are either not easy to scale with the ontology's size or have difficulty retrieving deeper compositional information [47]. Particularly important for reasoning methods are well-articulated inquiries. To solve this issue, we propose to embed larger information chunks of the FMEA data as vector representations to enable vector search.

Algorithm 1 Get vector embeddings from FMEA-KG

```
1:  procedure getVectorEmbeddings(FMEA_KG)
2:      ▷ vectorEmbeddings : set of vector embeddings
3:      for node ∈ failureEffects do
4:          connectedNodes ← DFS(node)
5:          ▷ chunk : string holding properties of connectedNodes
6:          for properties ∈ connectedNodes do
7:              chunk ← add(properties)
8:          end for
9:          vectorEmbeddings ← computeVectorEmbedding(chunk)
10:     end for
11:     return vectorEmbeddings
12: end procedure
```

Computing vector embeddings of FMEA information chunks. Following, we propose a method to create vector embeddings of FMEA data to allow vector search on the KG. To ensure meaningful information extraction from the vector search, vector embeddings must represent the entire FMEA graph. For this, we need to build text chunks from the KG that capture the underlying structure of the FMEA while not fragmenting the information into too small bits. For example, an inquiry such as *"What is the consequence of X on Y?"*, is not explicit and depends on the context X and Y. The results could differ greatly by a slight change with the context of the inquiry. Hence, it is essential to adopt an efficient approach that (i) encompasses all potential answers within the FMEA chunks, while simultaneously excluding irrelevant information, and (ii) scales with the size of the FMEA KG.

We propose traversing the FMEA KG by exploiting the tree-like structure of the FMEA KG (cf. Figure 2). For each *FailureEffect* node, we perform a depth-first search (DFS) to obtain the connecting nodes. This guarantees the collection of all branching nodes, such as when a *FailureEffect* node is related to multiple *FailureConsequences* nodes. Next, we compile a singular string representation of the entities traversed from the root node. With the generated text chunk, we compute the vector embedding utilizing any word-to-vector encoder. Algorithm 1 fully details the step for constructing the information chunks and computing the vector embeddings from the FMEA KG. Subsequently, we create a new node labeled *VectorEmbedding* that indexes the vector embedding and connects it to the corresponding *FailureEffect* node. In that *VectorEmbedding* node, we store the vector embeddings, the build text chunk, and the identifiers of the nodes, such as $processStepIds$.

3.2 KG RAG Framework

Subsequently, we propose the KG induced RAG framework for the retrieval of information from the FMEA KG proposed in Section 3.1. The concept of the framework is two fold and includes the retrieval of information with (i) the graph query language of the KG and (ii) vector search. The KG query language enables inquiries that search for specific results for which the object of the information is already known or for the

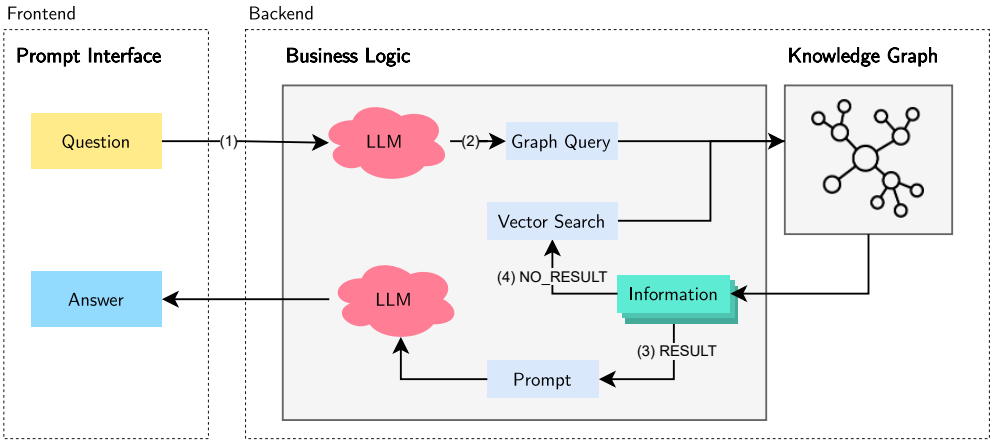

Fig. 3 Overview of the proposed KG RAG framework. The information is either retrieved utilizing the graph query language of the KG or vector search. The framework employs a LLM to generate the query and to serve the result.

retrieval of basic analytics. For example, *"In which process step does the failure effect X with the highest RPN occur?"* However, many questions do not provide sufficient context information for a query, as the context or object of the inquiry is unknown or implicitly articulated. To guarantee an information retrieval for an incomplete inquiry with missing context for a query language, we provide a vector search based on vector embeddings of the FMEA. Figure 3 summarizes the architecture of our proposed framework.

The framework operates through two pipelines: one for retrieving information from the FMEA KG, and another for serving the information to the user. Initially, we take the inquiry through a prompt interface and feed it into a LLM (cf. Fig. 3 (1)). The first action taken from the LLM is to generate a graph query, considering both the user's inquiry and the structure of the FMEA ontology (cf. Fig. 3 (2)). If a result of the graph query is obtained, the information retrieval process is complete, and the information is ready to be processed and presented to the user (cf. Fig. 3 (3)). If the graph query does not yield any results, the framework induces a vector search against the vector store (cf. Fig. 3 (4)).

This involves comparing the vector representation of the user's inquiry to the vectors from the database by calculating the cosine similarity between these two vectors. Cosine similarity is a measure defined by

$$\cos(\sigma) = \frac{A \cdot B}{\|A\| \cdot \|B\|} \tag{2}$$

where A represents the vector derived from the user's inquiry and B represents a vector from the store [48]. The similarity score ranges from -1 to 1, with -1 indicating

Table 1: Overview of *min*, *max* and *average* number of relationships for each label in the FMEA-KG. The KG has in total 3,107 nodes and 4,576 relationships.

Label	*min*(#relationships)	*max*(#relationships)	*avg*(#relationships)
FailureCause	2	11	3.45
FailureConsequence	1	68	15.7
FailureEffect	4	17	4.67
FailureMeasure	1	17	1.39
ProcessStep	2	71	20.30

completely opposite vectors, 0 orthogonal vectors, and 1 indicating very similar vectors. The framework selects the top k results with the closest similarity to 1. The parameter k depends on the user's inquiry. We set the default value to $k = 3$.

The top k results from the retrieval process are used as input to a LLM, which is prompted to generate a response that addresses the user's inquiry based solely on the information that has been retrieved from the FMEA KG. Thus, facilitating comprehension and mitigating hallucination.

4 Experiment and Discussion

Following, we evaluate the KG RAG framework tailored for FMEA QA. Our assessment focuses on the framework's capability of context retrieval. To facilitate user interaction with the KG RAG framework, we implement a simple chatbot interface, referred as FMEA chatbot. In Section 4.1, we detail the implementation details of the KG RAG framework and introduce the FMEA dataset. Subsequently, in Sections 4.2 and 4.3, we present the methodology for a human study and outline the results. In Section 4.4, we assess the capability of the chatbot to retrieve numerical data effectively by measuring the precision and recall of context. Finally, in Section 4.5, we discuss the results of our study and draw conclusions about the performance of the KG RAG framework in the context of FMEA.

4.1 Implementation Details

This Section details the specifics of our implementation and the FMEA data. For our results, we chose an actual FMEA dataset from a production of high-voltage systems (HVS) from prismatic battery cells. With up to 35% of the total production costs of an electric vehicle, HVS make up the most expensive component of the vehicle and, therefore, there is a great interest in the need for well-defined quality measurements [49]. The FMEA-KG comprises 3,107 nodes that encompass a wide range of different process steps, failure consequences, failure effects, failure causes, and failure measures. A total of 4,576 relations connect the nodes. An overview of the number of relationships *min*, *max* and *average* for each label in the FMEA-KG can be found in Table 4.1. The average number of relationships is between 1.39 (*FailureMeasure*) and 20.30 (*ProcessStep*).

For the construction of the multi-label property graph, we employ Neo4j [29], and make use of its Cypher graph query language to operate the graph data efficiently. The

LLM of our KG RAG framework is OpenAI's GPT-4 model with version 1106-Preview [42]. Vector embeddings are computed using the text-embedding-ada-002 model [50]. Another important aspect is the design of prompts used for running the LLM's inference jobs. The prompts ensure that the LLM generates precise and relevant responses, thereby enhancing the overall performance of the context retrieval. We implement the KG RAG framework's business logic as backend service with a simple REST-API and employ a simple chat interface for the study (cf. Figure 1)[1].

4.2 Study Methodology

Although FMEA is a complex expert topic, information retrieval is relevant to all technical professionals working in the shop floor who are involved in risk management. In our study, we engaged a group of 10 participants, including FMEA experts and non-experts to evaluate the effectiveness of our KG RAG framework. The participants were assigned the challenge of extracting information using the FMEA chatbot (cf. Section 4.1) and the original FMEA-Excel spreadsheet as a baseline. Assessing the performance of an information retrieval framework for FMEA presents unique challenges, as (i) there is no established baseline due to different FMEA approaches, (ii) the validity of the evaluation process in the study depends on the subject, and (iii) limited availability of domain experts. Nonetheless, in the context of FMEA, the use of Excel as a comparative baseline is grounded in its widely used tool to work with FMEA in the manufacturing sector, particularly for small and medium-sized companies.

Throughout the study, each participant was asked to complete three distinct tasks, each designed to test the retrieval of different types of FMEA information. The tasks were designed to simulate actual scenarios that professionals might encounter when dealing with FMEA on the shop floor. For example, "*In the process step X, there are faults due to Y. What prevention measure is provided for the elimination of the fault?*" For each task, the performance of the RAG framework and Excel was evaluated using five metrics: the correctness of the information retrieved, the usability of the interface, the relevance of the results to the task at hand, the completeness of the information provided, and the time taken to retrieve the necessary data. Employing these metrics allowed us, in addition to a qualitative assessment, to quantify and understand the advantages and potential drawbacks of the RAG approach.

We give the subjects the following definitions for correctness, usability, relevance, completeness, and retrieval time.

Correctness: Evaluates the degree of precision of the information retrieved by the subject and whether it corresponds to the task requirement. For example, are the results based on information from the FMEA database or hallucinated by the model.
Usability: Measures the ease of use with which participants interact with the system's interface. The interface is user-friendly, intuitive, and easy to navigate.
Relevance: Evaluates how closely the retrieved information aligns with the context of the question. Is there any unnecessary information for the task revealed?
Completeness: Indicates if the retrieved information provides a full and thorough

[1] For those interested in reproducing our results with example FMEA or exploring the framework further, the code for the RAG framework's backend service is available on GitHub at https://github.com/lukasbahr/kg-rag-fmea.

Table 2: We asked $n = 10$ participants to rate (1-5) each of three tasks according to the depicted metrics for the FMEA chatbot and Excel. The results show the accumulated mean value with the standard deviation.

	Correctnes	Usability	Relevance	Complete	Retrieval Time	Time Taken [min]
Excel	4.38 ± 0.80	2.10 ± 0.70	3.95 ± 1.02	3.86 ± 1.10	2.76 ± 0.77	01:51 ± 01:26
KG RAG	4.71 ± 0.71	4.71 ± 0.46	4.67 ± 0.91	4,38 ± 0.92	4.81 ± 0.40	01:19 ± 01:00

answer to the task. For example, for a task that asks for three failure consequences, the result should match the requirement.
Retrieval time: Measure the time and thus the efficiency of the retrieval process, in particular, how long it takes a subject to get the information.

After each task, we asked the participants to assign a score (1-5) for each metric, reflecting their notion of how well the task was achieved with the FMEA chatbot or the Excel spreadsheet. We also asked them to share their thoughts and comments during and after each task and stopped the time.

4.3 Results of the Study

The results of the study have been positive throughout and the FMEA chatbot was across board higher evaluated, indicating an improvement over Excel to access FMEA information. Table 2 summarizes the results. The correctness of the information retrieved from the FMEA chatbot is evaluated 7.53% higher compared to Excel. Completeness and relevance achieve 13.47% and 18.23% higher. Usability increased by 124.29%. Although the measured time the participants took for each task is on average 00:32 minutes (24.72%) shorter with the FMEA chatbot, their subjective perception of time differs with an evaluation of the retrieval time of 74.28% higher compared to Excel. The highest standard deviation for the chatbot appears for completeness (0.92) and corresponds to the thoughts and comments that we received from the participants. After the first task, the participants usually checked against the ground-truth FMEA data embodied in the Excel spreadsheet.

FMEA experts highlighted that the given context added to the answer helped to build trust in the chatbot's answer. The contexts provided sensible explanations, reduced the need for follow-up questions, and thus increased the relevance of the inquiry result. *"Context is great: gives a sensible explanation with more information. Saves to re-ask."* Other experts said that working with FMEA in Excel spreadsheets is not a pleasant task, for example, when seeking multiple explanations for a failure cause or gathering a general overview about the FMEA. One participant expressed: *"The chatbot approach makes the FMEA much more interesting for acquiring knowledge and for simply using. So far, the use of FMEA with Excel has not been used pleasantly, and you always needed help of the FMEA owners."*

All participants advertised the chatbot's ease of use. The ten participants were already familiar with chat-based LLMs, which helped them adapt to the novel setting. In particular, the ability to correctly interpret and respond to inquiries was highlighted. One user noted the proficiency of knowledge retrieval even though the input had vague language and, as non-FMEA expert, did not use the precise terminology typically

Table 3: Results on context recall and precision for a base RAG implementation and the proposed KG RAG framework.

	Context Recall	Context Precision
Base RAG	0.22	0.44
KG RAG	0.46	0.82

necessary for search queries in FMEA tools. *"There was a spelling mistake in the query, and it was relatively imprecise - I didn't use the precise term I was searching for but still got the correct answer."* Another participant stated: *"It's just cool that you don't have to be precise when describing and still get the right answer."*

Although all the participants had a positive impression of the proposed approach, some experts stated that a more thorough evaluation is necessary before deploying it on the shop floor. Another remark from FMEA experts was that it would be beneficial to integrate the chatbot into an existing FMEA tool.

4.4 Evaluation of Context Retrieval

Experts in FMEA often require precise numerical data extraction to improve risk mitigation strategies. For example, experts may need to determine the primary cause of failure within a particular process step, imposing questions such as *"What failure cause has the highest RPN for the process step X?"*. In addition, experts also employ basic data analysis, such as identifying the process step with the lowest, highest, or average RPN. In a FMEA Excel spreadsheet, that would entail a series of computational actions. The merits of our proposed RAG framework allow the retrieval of numerical data and basic statistical analysis of FMEA data, thanks to Cypher's graph query language [29].

To validate the effectiveness of our context retrieval method, we evaluate the RAG framework on a dataset comprising 30 entries, with questions and verified ground truth information, focusing on the extraction of numerical data. We involve FMEA experts to ensure that the validation dataset is unbiased and that the ground truth data are accurately defined. As a benchmark, we compare our RAG framework with a base RAG implementation that uses vector search for context retrieval [10]. The evaluation of the context retrieval is twofold, with the objective of verifying both context recall and precision on the validation dataset. Context recall (CR) measures the extent to which the retrieved context aligns with the ground truth data and is defined by

$$\mathrm{CR} = \frac{\#\text{ground truth attributable statements}}{\#\text{sentences in ground truth}} \tag{3}$$

for which # is the number of total appearance [10]. Context precision (CP) is evaluated on the basis of the ranking of the information within the context and its relevance. It

Table 4: Example results of the answers and retrieved contexts for the question, *"How many failure consequences with a S value of over 5 exist?"* and ground truth, *"There are 14 failures with a S score of over 5."* The questions are formulated in particular to test the retrieval of numerical information.

Model	Answer	Contexts
KG RAG	There are 14 failures with a S score of over 5.	["NumberFailureConsequencesWithSOver5: 14"]
Base RAG	There is one failure with a S-value of over 5.	["Process step: Form cell stacks, Failure consequence: Scrap cells/module, S: 6 (...)", "Process step: Form cell stacks, Failure consequence: Process failure/increase throughput times, S: 3 (...)", "Process step: Form cell stacks, Failure consequence: no particular effects, S:1 (...)"]

is calculated by

$$\mathrm{CP} = \frac{1}{\#\text{relevant information}} \sum_{m=1}^{n} \left(\frac{\#\text{relevant information till } m}{m} \times r_m \right) \tag{4}$$

where n denotes the total number of information in the context and r_m is the binary value whether the information to the m-th piece of information is true ($r_m = 1$) or false ($r_m = 0$) [51]. The dual assessment ensures that FMEA professionals can rely on the context retrieval system to provide accurate and precise contextual data.

Table 3 presents the results for CR and CP comparing the base RAG with the enhanced KG-induced RAG framework. The data indicates that our method yields an improvement, with a 209% increase in CR and a 186% increase in CP over the standard RAG performance. An examination of the retrieved context for the base RAG and the KG RAG framework is shown in Table 4, for an example inquiry, *"How many failure consequences with a S value of over 5 exist?"* The contexts illustrate that even though the baseline RAG approach correctly reasoned its answer from the incomplete context, it was unable to correctly retrieve the number of failure consequences from the FMEA data.

4.5 Discussion of the Results

The results constitute a first step towards a KG-enhanced RAG framework tailored for risk mitigation using FMEA data from manufacturing environments. Unlike conventional tabular structured FMEA access methods, such as Excel spreadsheets, the FMEA chatbot facilitates accelerated navigation through the data, allowing efficient identification of appropriate risk mitigation strategies. The KG RAG framework's ability to reason and comprehend vague queries is an improvement that enhances the accessibility and user-friendliness of FMEA. This will benefit both experts and non-experts in extracting insights from the data. The reduced information retrieval time helps practitioners on the shop floor by enabling agile decision-making and problem-solving.

Additionally, the results demonstrate that the integration of a graph query search with vector search enables basic analytical capabilities on datasets with mixed data types, such as FMEA, which improves CR and CP. This, along with the context visualization within the chat interface, fosters confidence in the KG RAG framework's capabilities. However, quality engineers are often in contact with various quality management systems. Based on the feedback from FMEA experts during our interviews, implementing a chatbot-like interface within an existing FMEA tool would presumably help in more complex error prevention cases.

Although, the study presents generally positive outcomes, the very nature of domain fitting of our approach brings threats to validity. The challenge lies in obtaining statistical significance across metrics such as correctness, usability, relevance, completeness, and retrieval time, given the size of our human study. In addition, conducting an automated evaluation proves challenging in the absence of the necessary expert knowledge, which complicates efforts to fully assess the performance of the KG RAG framework. Finally, more aspects of our approach need to be explored. In particular, we did not test with alternative LLMs for text generation, as access to other online language models is restricted by data protection regulations.

5 Conclusion and Perspectives

In this paper, we propose a KG RAG framework that synergies KG with LLM to enhance QA capabilities within the FMEA domain. For that, we come up with an ontology to model the FMEA data as a multi-label property KG. To compute the vector embeddings of the FMEA, we suggest an algorithm to capture the information from the KG. We then offer a solution for a novel RAG framework that is capable of retrieving FMEA information and allows for analytical inquiries.

The results of the human study suggest a promising direction for the retrieval and interpretation of the semantic content of the FMEA data, enabling a more nuanced and analytical approach to risk assessment. Additionally, the evaluation of the context retrieval demonstrated that the retrieval of numerical data is improved when augmented with the graph query interface of the KG. The findings indicate that a general integration of LLMs into the shop floor could advance quality strategies.

However, looking ahead many questions remain open. For one, the work's focus is on the retrieval of information, and thereby leaves open investigation to other LLMs for their effectiveness in generation and computing of vector embeddings. Moreover, the framework is suitable to be extended to support guided user input, which could significantly improve the data quality of the FMEA database. Further thinkable investigations are around expanding the knowledge base to encompass other quality management data, such as PDCA, or manufacturing data which could allow for a broader application of a RAG framework across different quality assurance contexts.

Acknowledgements

The authors would like to thank the engineers and co-workers on the shop floor who participated in the evaluation of the KG RAG framework. This research was co-funded by the Bavarian Ministry of Economic Affairs, Regional Development and Energy, the BayVFP's "Digitalization" funding line, the KIProQua project and the BMW Group.

References

[1] Colledani, M., Tolio, T., Yemane, A.: Production quality improvement during manufacturing systems ramp-up. CIRP Journal of Manufacturing Science and Technology **23**, 197–206 (2018) https://doi.org/10.1016/j.cirpj.2018.07.001

[2] Wehner, C., Kertel, M., Wewerka, J.: Interactive and Intelligent Root Cause Analysis in Manufacturing with Causal Bayesian Networks and Knowledge Graphs. In: 2023 IEEE 97th Vehicular Technology Conference (VTC2023-Spring), pp. 1–7. IEEE, Florence, Italy (2023). https://doi.org/10.1109/VTC2023-Spring57618.2023.10199563

[3] Eirich, J., Jäckle, D., Sedlmair, M., Wehner, C., Schmid, U., Bernard, J., Schreck, T.: ManuKnowVis: How to Support Different User Groups in Contextualizing and Leveraging Knowledge Repositories. IEEE Transactions on Visualization and Computer Graphics **29**(8), 3441–3457 (2023) https://doi.org/10.1109/TVCG.2023.3279857

[4] ISO/IEC: Risk management — Risk assessment techniques. Standard, International Organization for Standardization, Geneva, CH (June 2019). Volume: 2019 tex.key: ISO/IEC 31010:2019

[5] Schmitt, R., Pfeifer, T.: Qualitätsmanagement: Strategien, Methoden, Techniken, 5., überarbeitete auflage edn. Hanser, München (2015)

[6] Ozarin, N.W.: Bridging software and hardware FMEA in complex systems. In: 2013 Proceedings Annual Reliability and Maintainability Symposium (RAMS), pp. 1–6. IEEE, Orlando, FL (2013). https://doi.org/10.1109/RAMS.2013.6517739

[7] Carlson, C.S.: Effective FMEAs: Achieving Safe, Reliable, and Economical Products and Processes Using Failure Mode and Effects Analysis, 1st edn. Wiley, New Jersey (2012). https://doi.org/10.1002/9781118312575

[8] Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J.D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., Amodei, D.: Language Models are Few-Shot Learners. In: Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M.F., Lin, H. (eds.) Advances in Neural Information Processing Systems, vol. 33, pp. 1877–1901. Curran Associates, Inc., New York, NY, USA (2020)

[9] Pan, S., Luo, L., Wang, Y., Chen, C., Wang, J., Wu, X.: Unifying Large Language Models and Knowledge Graphs: A Roadmap. IEEE Transactions on Knowledge and Data Engineering, 1–20 (2024) https://doi.org/10.1109/TKDE.2024.3352100

[10] Shuster, K., Poff, S., Chen, M., Kiela, D., Weston, J.: Retrieval Augmentation Reduces Hallucination in Conversation. In: Findings of the Association for Computational Linguistics: EMNLP 2021, pp. 3784–3803. Association for Computational Linguistics, Punta Cana, Dominican Republic (2021). https://doi.org/10.18653/v1/2021.findings-emnlp.320

[11] Hogan, A., Blomqvist, E., Cochez, M., D'amato, C., Melo, G.D., Gutierrez, C., Kirrane, S., Gayo, J.E.L., Navigli, R., Neumaier, S., Ngomo, A.-C.N., Polleres, A., Rashid, S.M., Rula, A., Schmelzeisen, L., Sequeda, J., Staab, S., Zimmermann, A.: Knowledge Graphs. ACM Computing Surveys **54**(4), 71–17137 (2021) https://doi.org/10.1145/3447772

[12] Schramm, S., Wehner, C., Schmid, U.: Comprehensible Artificial Intelligence on Knowledge Graphs: A survey. Journal of Web Semantics **79**, 100806 (2023) https://doi.org/10.1016/j.websem.2023.100806

[13] Noy, N., Gao, Y., Jain, A., Narayanan, A., Patterson, A., Taylor, J.: Industry-scale knowledge graphs: lessons and challenges. Communications of the ACM **62**(8), 36–43 (2019) https://doi.org/10.1145/3331166

[14] Nigam, V.V., Paul, S., Agrawal, A.P., Bansal, R.: A Review Paper On The Application Of Knowledge Graph On Various Service Providing Platforms. In: 10th International Conference on Cloud Computing, Data Science & Engineering (Confluence), pp. 716–720 (2020). https://doi.org/10.1109/Confluence47617.2020.9058298

[15] Serrat, O.: The Five Whys Technique. In: Knowledge Solutions, pp. 307–310. Springer, Singapore (2017)

[16] Ruijters, E., Stoelinga, M.: Fault tree analysis: A survey of the state-of-the-art in modeling, analysis and tools. Computer Science Review **15-16**, 29–62 (2015) https://doi.org/10.1016/j.cosrev.2015.03.001

[17] Tay, K.M., Lim, C.P.: On the use of fuzzy inference techniques in assessment models: part II: industrial applications. Fuzzy Optimization and Decision Making **7**(3), 283–302 (2008) https://doi.org/10.1007/s10700-008-9037-y

[18] Gueorguiev, T., Kokalarov, M., Sakakushev, B.: Recent Trends in FMEA Methodology. In: 2020 7th International Conference on Energy Efficiency and Agricultural Engineering (EE&AE), pp. 1–4 (2020). https://doi.org/10.1109/EEAE49144.2020.9279101

[19] Spreafico, C., Russo, D., Rizzi, C.: A state-of-the-art review of FMEA/FMECA including patents. Computer Science Review **25**, 19–28 (2017) https://doi.org/10.1016/j.cosrev.2017.05.002

[20] Henshall, E., Campean, I.F., Rutter, B.: A Systems Approach to the Development and Use of FMEA in Complex Automotive Applications. SAE International Journal of Materials and Manufacturing **7**(2), 280–290 (2014) https://doi.org/10.4271/2014-01-0740

[21] Hunt, J.E., Pugh, D.R., Price, C.J.: Failure Mode Effects Analysis: A Practical Application of Functional Modeling. Applied Artificial Intelligence **9**(1), 33–44 (1995) https://doi.org/10.1080/08839519508945466

[22] Almannai, B., Greenough, R., Kay, J.: A decision support tool based on QFD and FMEA for the selection of manufacturing automation technologies. Robotics and Computer-Integrated Manufacturing **24**(4), 501–507 (2008) https://doi.org/10.1016/j.rcim.2007.07.002

[23] Mittal, S., Joshi, A., Finin, T.: Thinking, Fast and Slow: Combining Vector Spaces and Knowledge Graphs. arXiv (2017). https://doi.org/10.48550/ARXIV.1708.03310

[24] Schneider, E.W.: Course Modularization Applied: The Interface System and Its Implications For Sequence Control and Data Analysis. ERIC Clearinghouse, Alexandria, VA. (1973)

[25] Miller, G.A.: WordNet: A Lexical Database for English. Commun. ACM **38**(11), 39–41 (1995) https://doi.org/10.1145/219717.219748

[26] Auer, S., Lehmann, J., Cyganiak, R., Ives, Z.: DBpedia: A Nucleus for a Web of Open Data. In: The Semantic Web vol. 4825, pp. 722–735. Springer, Berlin, Heidelberg (2007)

[27] Bollacker, K., Evans, C., Paritosh, P., Sturge, T., Taylor, J.: Freebase: a collaboratively created graph database for structuring human knowledge. In: Proceedings of the 2008 ACM SIGMOD International Conference on Management Of data, pp. 1247–1250. ACM, Vancouver Canada (2008). https://doi.org/10.1145/1376616.1376746

[28] Singhal, A.: Introducing the Knowledge Graph: things, not strings. Visited on 01/01/2024 (2012). https://blog.google/products/search/introducing-knowledge-graph-things-not/

[29] Neo4j Graph Database & Analytics – The Leader in Graph Databases. Visisted on 03/01/2024. https://neo4j.com/

[30] Lin, T., Wang, Y., Liu, X., Qiu, X.: A survey of transformers. AI Open **3**, 111–132 (2022) https://doi.org/10.1016/j.aiopen.2022.10.001

[31] Holland, M., Chaudhari, K.: Large language model based agent for process planning of fiber composite structures. Manufacturing Letters **40** (2024) https:

//doi.org/10.1016/j.mfglet.2024.03.010

[32] Ji, Z., Lee, N., Frieske, R., Yu, T., Su, D., Xu, Y., Ishii, E., Bang, Y.J., Madotto, A., Fung, P.: Survey of Hallucination in Natural Language Generation. ACM Computing Surveys **55**(12), 1–38 (2023) https://doi.org/10.1145/3571730

[33] Rawte, V., Chakraborty, S., Pathak, A., Sarkar, A., Tonmoy, S.M.T.I., Chadha, A., Sheth, A., Das, A.: The Troubling Emergence of Hallucination in Large Language Models - An Extensive Definition, Quantification, and Prescriptive Remediations. In: Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, pp. 2541–2573. Association for Computational Linguistics, Singapore (2023). https://doi.org/10.18653/v1/2023.emnlp-main.155

[34] Danilevsky, M., Qian, K., Aharonov, R., Katsis, Y., Kawas, B., Sen, P.: A Survey of the State of Explainable AI for Natural Language Processing. In: Wong, K.-F., Knight, K., Wu, H. (eds.) Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing, pp. 447–459. Association for Computational Linguistics, Suzhou, China (2020). https://aclanthology.org/2020.aacl-main.46

[35] Wehner, C., Powlesland, F., Altakrouri, B., Schmid, U.: Explainable Online Lane Change Predictions on a Digital Twin with a Layer Normalized LSTM and Layer-wise Relevance Propagation. In: Advances and Trends in Artificial Intelligence. Theory and Practices in Artificial Intelligence vol. 13343, pp. 621–632. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-08530-7_52

[36] Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W.-t., Rocktäschel, T., Riedel, S., Kiela, D.: Retrieval-augmented generation for knowledge-intensive NLP tasks. In: Proceedings of the 34th International Conference on Neural Information Processing Systems. NIPS '20. Curran Associates Inc., New York, NY, USA (2020). https://doi.org/10.5555/3495724.3496517

[37] Li, H., Su, Y., Cai, D., Wang, Y., Liu, L.: A Survey on Retrieval-Augmented Text Generation. arXiv (2022). https://doi.org/10.48550/arXiv.2202.01110

[38] Gao, Y., Xiong, Y., Gao, X., Jia, K., Pan, J., Bi, Y., Dai, Y., Sun, J., Wang, M., Wang, H.: Retrieval-Augmented Generation for Large Language Models: A Survey (2023) https://doi.org/10.48550/ARXIV.2312.10997

[39] Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient Estimation of Word Representations in Vector Space (2013) https://doi.org/10.48550/ARXIV.1301.3781

[40] Devlin, J., Chang, M.-W., Lee, K., Toutanova, K.: BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In: Burstein, J., Doran,

C., Solorio, T. (eds.) Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pp. 4171–4186. Association for Computational Linguistics, Minneapolis, Minnesota (2019). https://doi.org/10.18653/v1/N19-1423

[41] Neelakantan, A., Xu, T., Puri, R., Radford, A., Han, J.M., Tworek, J., Yuan, Q., Tezak, N., Kim, J.W., Hallacy, C., Heidecke, J., Shyam, P., Power, B., Nekoul, T.E., Sastry, G., Krueger, G., Schnurr, D., Such, F.P., Hsu, K., Thompson, M., Khan, T., Sherbakov, T., Jang, J., Welinder, P., Weng, L.: Text and Code Embeddings by Contrastive Pre-Training (2022) https://doi.org/10.48550/ARXIV.2201.10005

[42] OpenAI: GPT-4 Technical Report. arXiv (2023). https://doi.org/10.48550/ARXIV.2303.08774

[43] Zhang, X., Bosselut, A., Yasunaga, M., Ren, H., Liang, P., Manning, C.D., Leskovec, J.: GreaseLM: Graph REASoning Enhanced Language Models for Question Answering. International Conference on Learning Representations (2023) https://doi.org/10.48550/ARXIV.2201.08860

[44] Jiang, J., Zhou, K., Zhao, W.X., Wen, J.-R.: UniKGQA: Unified Retrieval and Reasoning for Solving Multi-hop Question Answering Over Knowledge Graph. ICLR (2022) https://doi.org/10.48550/ARXIV.2212.00959

[45] Ahmed, T., Ghosh, S., Bansal, C., Zimmermann, T., Zhang, X., Rajmohan, S.: Recommending Root-Cause and Mitigation Steps for Cloud Incidents using Large Language Models. In: 2023 IEEE/ACM 45th International Conference on Software Engineering (ICSE), pp. 1737–1749 (2023). https://doi.org/10.1109/ICSE48619.2023.00149

[46] Tang, Y., Han, H., Yu, X., Zhao, J., Liu, G., Wei, L.: An Intelligent Question Answering System based on Power Knowledge Graph. In: 2021 IEEE Power & Energy Society General Meeting (PESGM), pp. 01–05 (2021). https://doi.org/10.1109/PESGM46819.2021.9638018

[47] Chen, X., Jia, S., Xiang, Y.: A review: Knowledge reasoning over knowledge graph. Expert Systems with Applications **141**, 112948 (2020) https://doi.org/10.1016/j.eswa.2019.112948

[48] Camacho-Collados, J., Pilehvar, M.T.: From Word To Sense Embeddings: A Survey on Vector Representations of Meaning. Journal of Artificial Intelligence Research **63**, 743–788 (2018) https://doi.org/10.1613/jair.1.11259

[49] Küpper, D.: The Future of Battery Production for Electric Vehicles. Visited on 11/01/2023 (2018). https://www.bcg.com/publications/2018/future-battery-production-electric-vehicles

[50] Greene, R., Sanders, T., Weng, L., Neelakantan, A.: New and improved embedding model. Visited on 03/10/2024 (2022). https://openai.com/blog/new-and-improved-embedding-model

[51] Salemi, A., Zamani, H.: Evaluating Retrieval Quality in Retrieval-Augmented Generation (2024) https://doi.org/10.48550/ARXIV.2404.13781